



Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología
Sección de Ingeniería Informática

Trabajo de Fin de Grado

Control de asistencia basado en reconocimiento robusto de
fiducials: aplicación para sistemas móviles
*Assistance Control System based on robust recognizing of fiducials: app
for mobile devices*

Adrián González Martín

La Laguna, 3 de julio de 2015

D. **Leopoldo Acosta Sánchez**, con N.I.F. 42.165.911-B Catedrático de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

D. **Antonio Luis Morell González**, con N.I.F. 78.705.025-Z Personal Docente e Investigador en Formación adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como cotutor

C E R T I F I C A (N)

Que la presente memoria titulada:

“Control de asistencia basado en reconocimiento robusto de fiducials: aplicación para sistemas móviles”

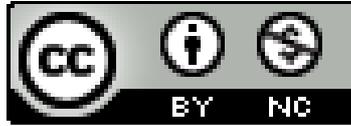
ha sido realizada bajo su dirección por D. **Adrián González Martín**, con N.I.F. 54.053.258-S.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 3 de julio de 2015

Agradecimientos

Me gustaría agradecer principalmente a mis tutores Leopoldo Acosta Sánchez y Antonio Luis Morell González por el apoyo y la confianza que me han dado durante el desarrollo del proyecto. También me gustaría agradecer a mi compañera Sara Martín Molina por haberme dado consejos útiles que me han permitido solventar problemas y a mis padres, porque sin su apoyo no podría haber llegado hasta aquí.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional.

Resumen

En este proyecto se ha desarrollado un sistema de control de asistencia para alumnos de la Universidad de La Laguna. Este sistema está compuesto principalmente por dos aplicaciones Android. La primera está orientada a los alumnos y permite generar un marcador fiducial o fiduciario [1]. Ese marcador, que se codifica y decodifica utilizando la librería ArUco [15], contiene el número de identificación del usuario o NIU. La segunda aplicación está orientada a los profesores de la universidad. Permite reconocer un marcador y extraer la información de cada alumno utilizando la cámara del dispositivo. El profesor debe comprobar si la información mostrada se corresponde con el alumno que le está mostrando el marcador y, en dicho caso, marcar su asistencia a clase.

Se ha desarrollado también una aplicación de escritorio en Java que permite generar un fichero con los diferentes alumnos de una asignatura. De esta forma, cada profesor podrá generar diferentes ficheros en los que se encuentre la información de sus alumnos para varias asignaturas. Algunos de los datos que se pueden almacenar para cada alumno son su nombre, sus apellidos, su NIU y su foto.

Tanto la aplicación del profesor como la aplicación de escritorio utilizan una base de datos H2[13]. Se ha seleccionado este gestor de bases de datos porque cuenta con compatibilidad para aplicaciones Android y porque permite exportar e importar tablas desde un fichero CSV[2]. Esta última función ha sido muy útil para importar los alumnos de una asignatura en la aplicación del profesor y para exportar las listas con las faltas de los alumnos.

Palabras clave: *control de asistencia, marcador fiducial, ArUco, aplicación escritorio, aplicación dispositivos móviles, Android, Java, base de datos H2*

Abstract

In this project we have developed an assistance control system for the students of the University of La Laguna. This system is based in two Android applications. The first one is oriented to the students which can be used to generate a custom fiducial marker [1]. This marker can be encoded and decoded using ArUco[15] library. It contains the user identification number (NIU). The second application is meant to be used by the teachers of the university. It can recognize a marker and extract all the information about the student through the device's camera. The teacher must check if the information showed corresponds to the student who is showing the marker. If everything is ok, he can set his assistance to class.

We have also developed a Java desktop application with which the teacher can generate a file with all the students of a subject. This way, each teacher can generate different files for several subjects where he can store the information about his students. He can store the name of the student, his surname, his NIU and his photo.

Both teacher and desktop applications use a H2 database [13]. We have selected this database engine because it has compatibility with Android applications and we can use it to import or export tables from a CSV file [2]. This feature has been very useful for importing the students attending a subject in the teacher's application. It also has been useful for exporting the lists with the absences of the students.

Keywords: *assistance control, fiducial marker, ArUco, desktop application, mobile device application, Android, Java, H2 database*

Índice general

Capítulo 1 Introducción.....	1
Capítulo 2 Antecedentes.....	2
Capítulo 3 Estado del arte.....	4
3.1 Códigos de Barras.....	5
3.2 Marcadores fiduciarios.....	6
3.3 Marcadores ArUco.....	8
Capítulo 4 Objetivos.....	10
Capítulo 5 Fases y desarrollo del proyecto.....	12
5.1 Fases.....	12
5.1.1 Fase de documentación.....	12
5.1.2 Fase de diseño.....	13
5.1.3 Fase de desarrollo.....	17
5.1.4 Fase de testeo.....	22
5.2 Funcionamiento de las aplicaciones Android.....	23
5.2.1 Funcionamiento Aplicación Alumno.....	23
5.2.2 Funcionamiento Aplicación Profesor.....	24
Capítulo 6 Conclusiones y líneas futuras.....	28
Capítulo 7 Summary and Conclusions.....	30
Capítulo 8 Presupuesto.....	32
8.1 Presupuesto por fases.....	32
Bibliografía y referencias.....	33

Índice de Figuras

Figura 2.1. Dispositivo de gestión docente.....	2
Figura 3. Uso de la aplicación Vía-Móvil.....	4
Figura 3.1. De izquierda a derecha: código de barras estándar, DataMatrix, MaxiCode y QR.....	6
Figura 3.2.1. Marcador circular.....	7
Figura 3.2.2. De izquierda a derecha: ARTag, ARToolKit y BinARyID.....	7
Figura 3.3. Marcador generado por ArUco.....	8
Figura 5.1.2.1. Prototipos realizados en MarvelApp.....	14
Figura 5.1.2.2. Diseño de marcador propuesto.....	15
Figura 5.1.2.3. Ejemplo de codificación.....	16
Figura 5.1.2.4. Correspondencia de submarcadores en el ejemplo.....	17
Figura 5.1.3. Interfaz de la aplicación Java de escritorio.....	19
Figura 5.1.4. Captura de la web creada.....	23
Figura 5.2.1. Capturas de la aplicación del alumno.....	24
Figura 5.2.2.1. Pantalla de inicio de sesión y principal.....	25
Figura 5.2.2.2. Pantalla de cámara y pantalla con la información de un alumno reconocido.....	26

Índice de tablas

Tabla 5.1: Cronograma con las fases realizadas durante el desarrollo.....12

Tabla 8.1: Presupuesto.....33

Capítulo 1

Introducción

Se ha propuesto el desarrollo de un sistema que permita realizar el control de asistencia a clase de alumnos de la Universidad de La Laguna utilizando un marcador fiducial o fiduciario [1]. Para cumplir este objetivo se ha desarrollado una aplicación de escritorio en Java y dos aplicaciones Android, una para el profesor y otra para el alumno. La aplicación de escritorio le permite al profesor crear una lista con los alumnos de una asignatura y exportarla a un fichero CSV [2]. Una vez creado ese fichero, debe subirlo a su cuenta de Google Drive e importarlo desde la aplicación Android.

Por otro lado, cada alumno cuenta también con una aplicación que genera marcadores en base al número identificador del alumno (NIU) que se introduzca. De este modo, cada alumno genera su propio marcador y el profesor, utilizando su aplicación, reconoce el marcador de cada alumno y marca su asistencia a clase.

Para llevar a cabo este proyecto, se ha evaluado el tipo de información que se debe codificar y el tipo de marcador fiduciario que mejor se adapta a las necesidades del sistema de control de asistencia.

Capítulo 2

Antecedentes

Los profesores tienen diferentes modos de realizar el seguimiento del alumnado. El más básico que podemos considerar es el de utilizar una lista impresa en papel donde ir marcando los alumnos que asisten a clase. Partiendo de este modelo, también existe una versión más moderna que consiste en pasar lista utilizando un ordenador con una hoja de cálculo. Además, existen centros educativos donde desde hace varios años los profesores utilizan un dispositivo de gestión docente denominado SGD. Gracias al mismo pueden pasar lista y amonestar a alumnos que muestren un mal comportamiento.



Figura 2.1. Dispositivo de gestión docente

Este dispositivo puede ser sustituido por una aplicación para Android denominada SGD App [3] que proporciona una interfaz mucho más simple al profesor.

Si nos centramos en el uso de aplicaciones móviles, podemos encontrar un gran número de ellas que permiten pasar lista utilizando un dispositivo Android o IOS. Algunos ejemplos de estas aplicaciones son Asistente Docente [4], Cuaderno del Profesor [5] o Cuaderno del Profesor Additio [6] para Android y Teacher Kit [7], iTeacherBook [8] o iDoceo [9] para IOS. Sin embargo, ninguna de estas aplicaciones utiliza marcadores para identificar a los alumnos. Dicho proceso de identificación queda en manos del profesor teniendo que ir nombrando a cada alumno para comprobar que asiste a clase y marcar su asistencia.

La principal diferencia de todos estos sistemas con el que se desarrolla en este proyecto es que para realizar la identificación de alumnos se utiliza una serie de marcadores únicos para cada uno de ellos. De esta manera, el profesor sólo tiene que escanear con su aplicación cada marcador para obtener automáticamente la ficha de cada alumno. Esto agiliza la identificación del alumno por lo que el proceso de pasar lista termina realizándose en menos tiempo. Además, dado que la aplicación le muestra al profesor una foto del alumno identificado, puede evitar que un alumno se haga pasar por otro para evitar la falta de asistencia.

Capítulo 3

Estado del arte

Hoy en día existen equipos de trabajo alrededor del mundo que están desarrollando proyectos en los que utilizan diferentes tipos de marcadores para desarrollar proyectos innovadores.

Este proyecto de fin de grado está basado en el uso de marcadores fiduciaros. Estos nos permiten localizar su proyección en una imagen que se obtiene de una cámara. Sin embargo, existen otros proyectos en los que no interesa obtener la proyección del marcador sino cierta información que tenga codificada. Para este tipo de proyectos no se utilizan los fiduciaros sino códigos de barras que pueden ser unidimensionales o bidimensionales.

En general, todos estos marcadores se utilizan en campos tan diferentes como la realidad aumentada para mostrar objetos tridimensionales en el campo de la educación, como medio de codificación de un enlace o información para publicitar diferentes productos o incluso como soporte que facilite el pago por Internet.

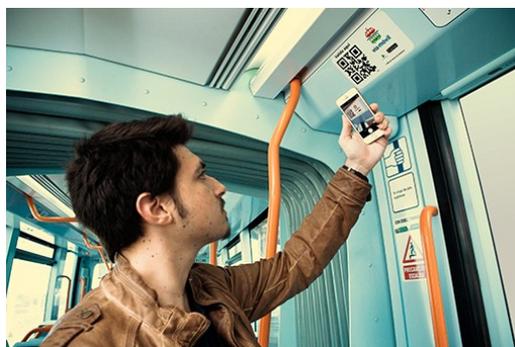


Figura 3. Uso de la aplicación Vía-Móvil

En relación a este último ejemplo, existe en Tenerife una aplicación denominada Vía-Móvil [10] que permite utilizar la guagua o el tranvía y pagar cada viaje escaneando el código QR que posee el vehículo en el que se va a viajar.

Esta aplicación se basa en la utilización de códigos QR que se pueden encontrar en los diferentes vehículos de transporte. Estos códigos no son los únicos que se podían haber utilizado ya que hoy en día existe una gran variedad de marcadores con diferentes propiedades, formas y tamaños. Cada uno de ellos tiene más utilidad y fiabilidad en un campo que en otro. Por ejemplo, a la hora de codificar el código de un producto, tiene más sentido utilizar un marcador bidimensional de barras verticales que un marcador fiduciario que este diseñado simplemente para obtener su posición en el espacio. Además, la utilidad y fiabilidad de cada marcador depende de diferentes aspectos, como pueden ser el promedio de falsos reconocimientos correctos o incorrectos, su capacidad de adaptación a diferentes condiciones lumínicas, la posibilidad de reconocerlos desde diferentes perspectivas y distancias o el número de marcadores diferentes de un mismo tipo que se pueden reconocer.

Dentro de la gran variedad de marcadores existentes hoy en día, podemos diferenciar dos grandes grupos: los marcadores basados en códigos de barras para codificar información y los marcadores fiduciarios que nos permiten obtener su ubicación en el espacio.

3.1 Códigos de Barras

Este grupo de marcadores se divide en dos categorías: los marcadores que se leen en una dimensión y que se leen en dos dimensiones. Los del primer tipo son los códigos de barras estándar que se utilizan desde hace muchos años en la identificación de productos. Para poder extraer la información que contienen se utiliza un escáner láser mientras que para extraer la información de los marcadores del segundo tipo se utiliza una cámara. Dentro de los de

segundo tipo encontramos el denominado DataMatrix, el MaxiCode y el más conocido QR. Todos estos marcadores se crean utilizando sólo dos tonos (normalmente blanco y negro) ya que, de esta forma, es más sencillo analizar y decodificar la imagen que contiene el marcador. Permiten codificar información pero no admiten modificaciones en la perspectiva de la cámara que los capta ni que el objetivo de la cámara se aleje mucho del marcador. Estos motivos, junto con que no son buenos marcadores para obtener su posición en el espacio, explican porque no son utilizados en sistemas de marcadores fiduciaros.



Figura 3.1. De izquierda a derecha: código de barras estándar, DataMatrix, MaxiCode y QR.

3.2 Marcadores fiduciaros

Estos marcadores son los utilizados cuando deseamos capturarlos con una cámara y obtener su posición en el espacio. Existen, al igual que en los explicados anteriormente, diferentes tipos de marcadores.

- Marcadores basados en puntos: estos marcadores son los más simples que podemos utilizar. Como su propio nombre indica, al analizar una imagen que contiene este marcador se trata de localizar los diferentes puntos que forman el marcador. Estos puntos pueden ser representados utilizando Leds, bombillas o simplemente dibujándolos de manera que destaquen lo suficiente para ser reconocidos.
- Marcadores circulares: estos marcadores están compuestos por una serie de sectores circulares o por anillos concéntricos. Como en el caso anterior, estos marcadores se utilizan para proporcionar la ubicación de un punto que se encuentra, normalmente, en el centro de esos círculos

concéntricos.

Además, este tipo de marcadores destaca por la precisión con la que son capaces de localizar dicho punto.

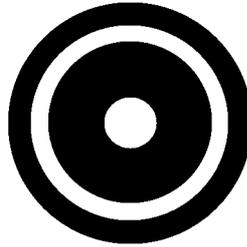


Figura 3.2.1. Marcador circular

- **Marcadores cuadriláteros:** este tipo de marcadores proporcionan cuatro puntos, las cuatro esquinas del cuadrilátero, que son muy útiles para calcular la posición de los mismos y para decodificarlos. Estos marcadores suelen ir acompañados de diccionarios de marcadores en los que se define, para cada tipo, el conjunto de marcadores que es posible reconocer. Si ese diccionario es muy grande, es posible que haya marcadores que se parezcan y eso produzca reconocimientos erróneos. Además, la forma en la que esté diseñado el algoritmo que detecta los marcadores también producirá una mayor o menor tasa de error durante el reconocimiento. Dentro de este tipo de marcadores podemos encontrar el sistema de marcadores fiduciario denominado ARTag, el ARToolKit o el BinARyID.



Figura 3.2.2. De izquierda a derecha: ARTag, ARToolKit y BinARyID.

3.3 Marcadores ArUco

Después de analizar los diferentes sistemas disponibles para codificar marcadores, se ha optado por utilizar la librería ArUco [15], en concreto su versión para Android, tanto para generar los distintos marcadores como para reconocerlos. Esta librería ha sido creada por algunos profesores del departamento de Computación y Análisis Numérico de la Universidad de Córdoba, entre los que se encuentran S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, M.J. Marín-Jiménez [33]. Se han creado diferentes versiones de la librería en Java, Python o JavaScript y respecto a otros sistemas, este cuenta con un diccionario compuesto por 1024 marcadores fiduciaros diferentes. Destaca por encima de los otros sistemas por su robustez y capacidad de detección de marcadores desde diferentes perspectivas y distancias. Además, una vez que se reconoce un marcador del diccionario, es capaz de detectar su identificador (un valor de 0 a 1023), su posición en el espacio y su rotación respecto a la posición original.

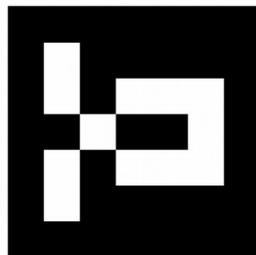


Figura 3.3. Marcador generado por ArUco.

Como se puede ver en la figura 3.3, los marcadores generados por ArUco están compuestos por un borde exterior de color negro. Dicho borde se utiliza para mejorar el reconocimiento del marcador. Dentro del mismo, encontramos una matriz de 5x5 cuadrados blancos o negros que son los que se modifican para generar distintos marcadores.

En este proyecto, la posición en el espacio del marcador es un aspecto irrelevante pero, como se verá más adelante, tanto el valor del identificador de cada marcador como el número de rotaciones de 90° que se detecten han

sido aspectos fundamentales para codificar los números de identificación de los alumnos de la Universidad de La Laguna.

A continuación, para explicar el procedimiento realizado para detectar un marcador, se presenta el algoritmo que utiliza la librería ArUco.

Algoritmo: Detectar marcadores ArUco

Data: Matriz OpenCv con información del frame capturado

Result: Vector con los marcadores detectados

- Convertir imagen en blanco y negro en base a un valor preestablecido "t" (threshold) para obtener los contornos. Seguir luego trabajando sobre la imagen original.
- Borrar contornos no deseados eliminando aquellos con pocos puntos.
- Detectar polígonos a través de los contornos y mantener aquellos cóncavos con 4 esquinas.
- Ordenar las esquinas en contra de las agujas del reloj.
- Eliminar los polígonos muy cercanos para evitar duplicidades.
- Detectar los marcadores:
 - Eliminar perspectiva para lograr una vista frontal utilizando homografía.
 - Convertir de nuevo la imagen en blanco y negro utilizando el método Otsu [20].
 - Identificar el código interno comprobando si se localiza el borde negro del marcador y luego leyendo la matriz de 5x5 celdas internas que codifican el identificador.
- Si se detectan marcadores válidos, hacer un refinamiento de las esquinas.
- Si se proporcionan los parámetros de la cámara, calcular la posición del marcador respecto a la cámara.

Como se puede ver, esta librería procesa la imagen que contiene el marcador eliminando aquellos elementos que entorpezcan el reconocimiento. Una vez obtenido el código interno se busca en el diccionario de marcadores y si existe se obtiene un identificador con un valor entre 0 y 1023. Como se ha comentado, será ese valor el que se utilice para codificar parte de NIU.

Capítulo 4

Objetivos

Para desarrollar este proyecto se han tenido en cuenta principalmente cinco objetivos.

- Documentación: este primer objetivo ha consistido en analizar los diferentes marcadores disponibles para codificar información. En dicho análisis se ha tenido en cuenta el tipo de información que es capaz de codificar cada uno de ellos, el dispositivo necesario para reconocer los marcadores (si bastaba una cámara o se necesitaba un escáner láser), la robustez de cada sistema, la capacidad para detectar marcadores con diferentes ángulos o inclinaciones de la cámara y la tolerancia a los cambios de luminosidad o distancias desde la cámara al marcador.
- Elección y diseño del marcador: este objetivo ha consistido en utilizar la información recopilada para elegir un tipo de marcador y después diseñar el marcador final que utilizará el alumno.
- Creación de la aplicación de escritorio que genera la base de datos con los alumnos: este objetivo ha consistido en crear una aplicación con interfaz propia que gestione una base de datos con alumnos. Para ello se han tenido que consultar diferentes sistemas gestores de bases de datos. Uno de los factores que se ha tenido en cuenta durante este proceso de búsqueda es que el sistema gestor seleccionado debía funcionar tanto en Windows como en Android.

- Creación de la aplicación para Android que genera marcadores: este objetivo ha consistido en el desarrollo de la aplicación que utilizarán los alumnos. Este objetivo y el último han sido los más tediosos al tener que utilizar librerías externas en Android para poder tanto generar como detectar los marcadores.
- Creación de la aplicación para Android que reconoce marcadores y establece la asistencia a clase: este último objetivo ha consistido en desarrollar la aplicación que utilizarán los profesores. Esta aplicación debía primero mostrar una pantalla inicial de logueo y después debía permitir utilizar la cámara del dispositivo para capturar el marcador. Una vez se reconociera un marcador, debía mostrar la información del alumno vinculado y permitir indicar que ese alumno acudió a clase. Además, otro de los aspectos que debía cumplir esta aplicación era, por un lado, permitir importar las listas con los alumnos desde Google Drive y, por otro lado, permitir exportar desde su dispositivo Android a Google Drive las mismas listas pero con los resultados de asistencia de los diferentes días que el profesor hubiera realizado el control de asistencia.

Capítulo 5

Fases y desarrollo del proyecto

El desarrollo del proyecto se ha llevado a cabo durante una serie de fases que han permitido la correcta finalización del mismo. La duración de cada una de estas fases ha sido la siguiente.

Fase	Período	Duración
Documentación	Del 16/02/2015 al 13/03/2015	4 semanas
Diseño	Del 16/03/2015 al 17/04/2015	5 semanas
Desarrollo	Del 20/04/2015 al 12/06/2015	8 semanas
Testeo	Del 15/06/2015 al 26/06/2015	2 semanas

Tabla 5.1: Cronograma con las fases realizadas durante el desarrollo..

5.1 Fases

5.1.1 Fase de documentación

Durante esta fase, primero se ha realizado una búsqueda de sistemas existentes que se utilicen para realizar el control de asistencia al alumnado. Gracias a esta búsqueda se localizó el sistema SGD, del que se habla brevemente en el capítulo 2. Este sistema de gestión docente es uno de los más utilizados en los centros educativos de Canarias. Cuenta con una aplicación tanto para IOS como para Android. Lo interesante de esa

aplicación, denominada SGDApp [3], es que permite que el profesor pase lista en clase como en una de las aplicaciones que se ha desarrollado en este proyecto. Sin embargo, el proceso sigue siendo el tradicional de ir nombrando a cada alumno en voz alta para luego marcar que asiste si levanta la mano. La aplicación desarrollada en este proyecto para el profesor utiliza la cámara para identificar automáticamente al alumno.

Además de SDGApp, se localizaron otras aplicaciones para dispositivos móviles que funcionaban de manera similar. La gran diferencia es que estas otras aplicaciones no están tan extendidas ni son tan usadas como lo es SGDApp.

Una vez que se había comprobado cómo funcionaban otras aplicaciones con un propósito similar, se pasó a buscar información sobre los diferentes marcadores y formas de codificar información en imágenes disponibles. Para ello se consultaron diversos artículos, siendo los más importantes *Automatic generation and detection of highly reliable fiducial markers under occlusion* [11] y *Designing Highly Reliable Fiducial Markers* [12]. El primero de los artículos ha sido crucial para desarrollar este proyecto puesto que es el que describe como se crean y detectan los marcadores de ArUco que son los que se han terminado utilizando.

También se ha buscado documentación sobre las diferentes alternativas disponibles para implementar una base de datos en la aplicación de escritorio que genera las listas de alumnos para una asignatura. Finalmente, se decidió implementar una base de datos H2 [13] porque era compatible con Android y permitía agilizar el proceso de desarrollo.

Por último, se ha buscado documentación de cómo crear y modificar aplicaciones Android. Esta información ha sido muy útil para realizar el desarrollo de las aplicaciones para el profesor y el alumno.

5.1.2 Fase de diseño

Durante esta fase, se han realizado los prototipos de las aplicaciones Android y el diseño del marcador que codifica el NIU del alumno.

En cuanto a los prototipos de las aplicaciones, se ha utilizado prototipado en papel combinado con la web MarvelApp [14]. Esta web permite transformar el diseño en papel en una aplicación dinámica. Para ello da la posibilidad de añadir vínculos de manera que al pulsar sobre un botón dibujado te lleve a otra de las pantallas que se han diseñado. De esa manera se han podido crear los prototipos de las aplicaciones de forma fácil y cómoda.



Figura 5.1.2.1. Prototipos realizados en MarvelApp

Por otro lado, en esta fase también se ha realizado el diseño del marcador que utilizará cada alumno. El primer paso dado ha sido elegir el sistema de marcadores que más se ajuste a las necesidades que presenta este proyecto. De toda la información de un alumno de la ULL que se podría codificar, se ha optado por utilizar sólo el NIU, es decir, un número entero compuesto por 10 dígitos. Para codificarlo se ha elegido utilizar una combinación de cuatro marcadores generados por la librería ArUco [15]. Se

ha seleccionado esta librería por encima de otras por su robustez, sencillez de uso e integración con Android.

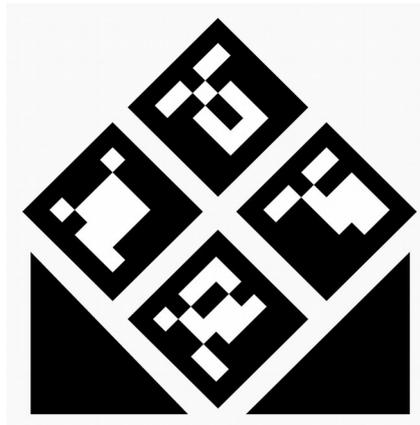


Figura 5.1.2.2. Diseño de marcador propuesto.

La forma de codificar el NIU es creando un nuevo marcador compuesto por cuatro submarcadores generados por ArUco y rotados 45 grados como se ve en la imagen superior. El NIU se divide en 5 grupos de dos dígitos cada uno, de manera si el NIU es 0123456789, el valor del primer grupo (01) se codifica con rotaciones de los submarcadores. Estas rotaciones son siempre de 90 grados y en contra de las agujas del reloj. Los otros cuatro grupos (23, 45, 67 y 89) se codifican con los valores de los identificadores de cada uno de los cuatro submarcadores.

Para codificar el primer grupo con rotaciones se utiliza un número binario de 8 dígitos que nos permite representar los valores del 0 al 99 necesarios y que se divide en cuatro partes de dos dígitos. Cada una de estas partes está vinculada a uno de los submarcadores. Para decodificar la rotación, se analiza cada submarcador de manera que, si no se ha rotado se asigna el valor binario '00', si se detecta una rotación se asigna '01', si se detectan dos rotaciones se asigna '10' y si se detectan tres rotaciones se asigna '11'. Al hacer este proceso con los cuatro submarcadores podemos obtener el número binario de ocho dígitos que transformado a decimal nos indica los dos primeros dígitos del NIU.

Para codificar los cuatro grupos restantes del NIU se utiliza el identificador de cada submarcador (valor entre 0 y 1023). Si estamos codificando el primer grupo el identificador será 1XY, siendo XY los dos dígitos de ese grupo. Los otros tres grupos se forman de manera similar pero modificando el primer valor por 2, 3 y 4. Estos dígitos iniciales son muy necesarios porque nos permiten facilitar la identificación de los dígitos de cada grupo. Cuando la librería ArUco detecta una imagen en la que aparecen cuatro marcadores, devuelve un vector con los identificadores de los marcadores detectados ordenados de menor a mayor. Gracias a estos dígitos iniciales el único paso que hay que dar para obtener el valor original que está codificado es eliminar dicho dígito. Si no se añadiera ese dígito no podríamos saber el grupo que se corresponde con cada elemento del vector retornado por la librería.

Para entender mejor el proceso de creación de un marcador, veamos un ejemplo de codificación para el NIU "0100536836":

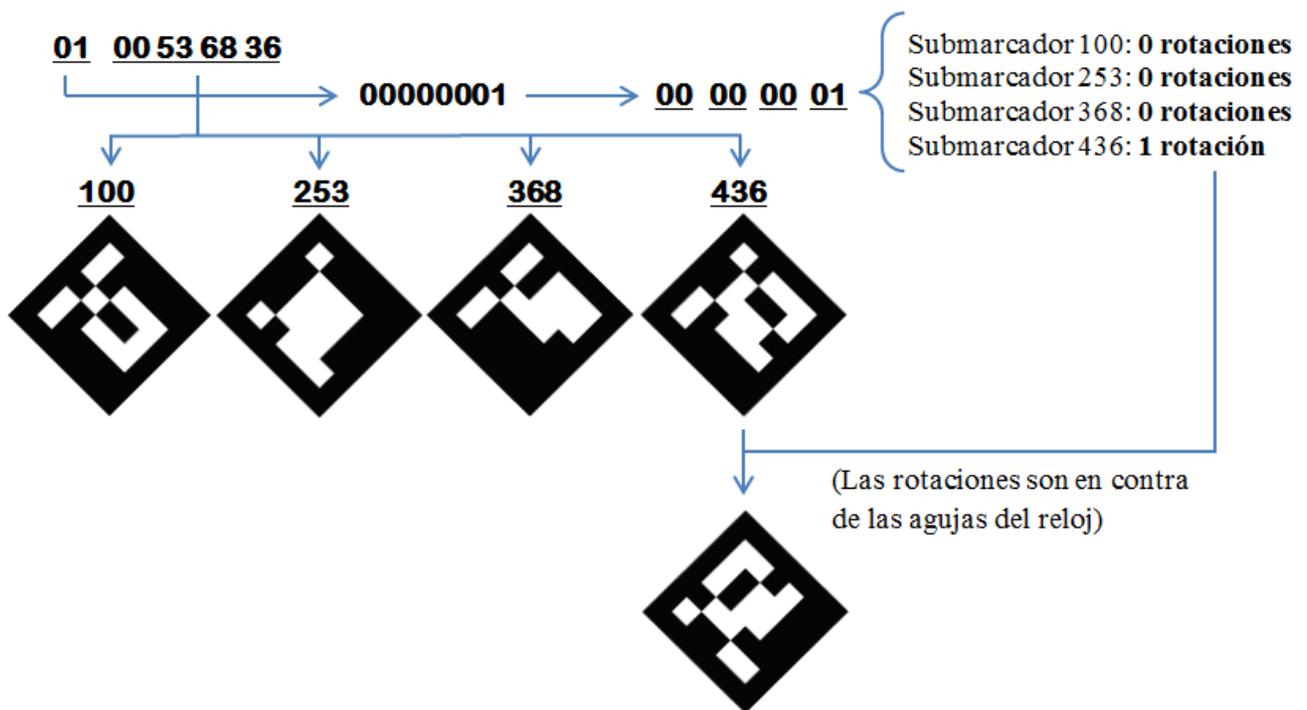


Figura 5.1.2.3. Ejemplo de codificación.

Continuando con el ejemplo, una vez que se tenemos cada submarcador sólo nos queda colocarlos en su posición correspondiente y dibujar los triángulos inferiores para obtener el marcador que identifica al alumno con NIU “0100536836”.

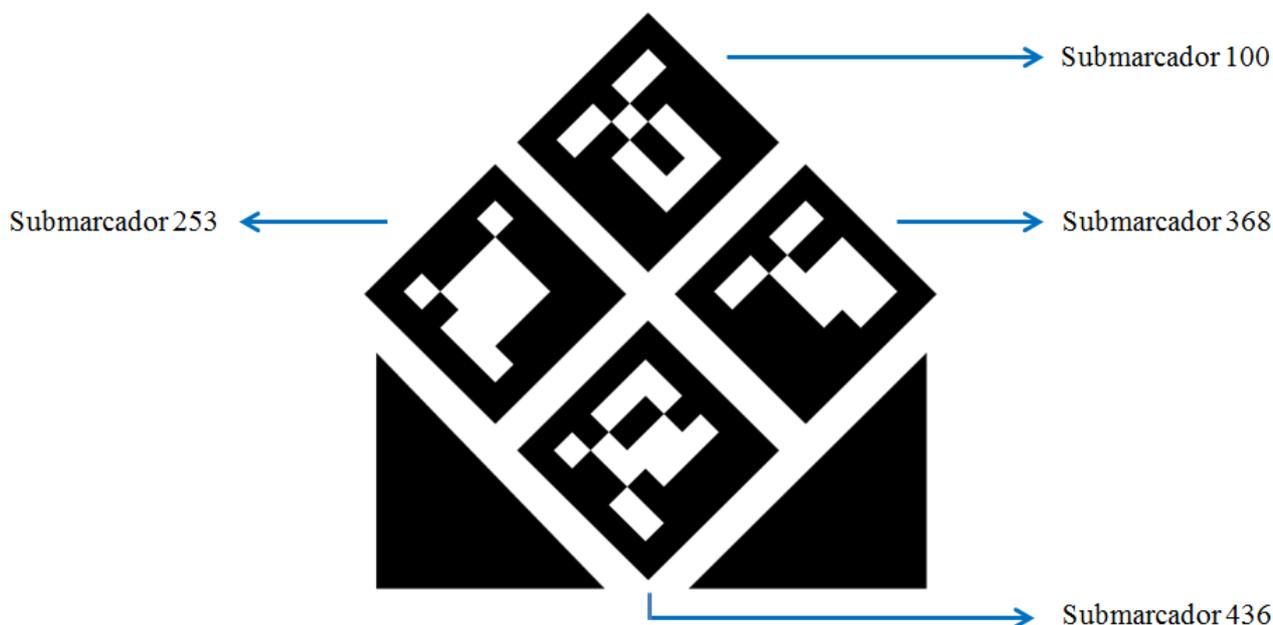


Figura 5.1.2.4. Correspondencia de submarcadores en el ejemplo.

Otra de las alternativas que se planteó cuando se diseñó el marcador era añadir cinco submarcadores y codificar todo el NIU con los identificadores. Sin embargo, esta alternativa fue descartada porque al poner cinco submarcadores se necesitaba más espacio y había que disminuir el tamaño de los otros submarcadores. Esto afectaría a la fase de detección de marcadores teniendo que acercarse más la cámara al marcador para lograr el mismo nivel de detalle que se tenía con cuatro submarcadores a una determinada distancia.

5.1.3 Fase de desarrollo

El desarrollo, tanto de la aplicación de escritorio como de las dos aplicaciones Android, se ha realizado en un ordenador con sistema operativo Ubuntu 14.04 LTS. Los elementos más importantes que se han utilizado

durante el desarrollo son los siguientes:

- Base de datos H2: se trata de una base de datos SQL en Java con licencia de código abierto y software libre. Es multiplataforma y permite acceder a la base de datos lanzando código SQL directamente. En este proyecto se ha utilizado la última versión disponible (1.4.187) que fue lanzada en abril del presente año.
- Librería ArUco: se trata de una librería que permite generar marcadores e identificarlos. Se utiliza en aplicaciones de realidad aumentada y es multiplataforma. Cuenta con una licencia de tipo BSD (software libre permisiva) y en este proyecto se ha utilizado la versión para Android. Más información en el apartado 3.3.
- Librería OpenCv: se trata de una librería para visión artificial de la que depende ArUco. Está programada en C/C++ para mejorar su eficiencia y cuenta con una licencia BSD (software libre permisiva). Permite realizar diversas funciones como reconocimiento de objetos, detección de movimiento o visión robótica. Aunque la última versión estable es la 3.0.0 (lanzada el 4 de junio de 2015), en este proyecto se ha utilizado la versión 2.4.10.
- API de Google Drive para Android: se trata un API de Android que se incluye en la librería de servicios de Google Play. En este proyecto se ha utilizado la versión 7.3.0. que proporciona la funcionalidad de descarga y subida de ficheros a una cuenta de Google Drive.

Desarrollo de la aplicación de escritorio

Primero ha sido creada la aplicación Java de escritorio que utiliza el profesor para crear la lista de alumnos de su asignatura. Esta aplicación se ha creado utilizando el entorno el entorno de desarrollo Eclipse Luna (versión

4.4.2) y como se ha comentado anteriormente utiliza una base de datos H2 para gestionar los alumnos.

El profesor tiene la posibilidad de introducir para cada alumno su nombre, apellidos, NIU y foto identificativa. Para poder almacenar la foto en la base de datos se codifica utilizando Base64 [16]. Una vez rellenados esos datos se debe pulsar sobre añadir alumno para almacenarlo. Si el profesor quiere eliminar algún alumno puede hacerlo pulsando sobre el botón correspondiente e introduciendo el NIU de dicho alumno. También tiene la posibilidad de borrar la base de datos por completo si así lo desea.

Cuando el profesor termina de introducir los alumnos, puede exportar la base de datos que ha creado a un fichero CSV. Este fichero es de vital importancia puesto que es el que se utiliza desde la aplicación Android del profesor para cargar la información de los alumnos a una asignatura. Por ese motivo, el profesor debe subir dicho fichero a su cuenta de Google Drive. La aplicación del dispositivo móvil cuenta con una función denominada “importar base de datos” que se conecta a la cuenta de Google Drive y muestra todos los ficheros con formato CSV que contiene. De esa manera el profesor puede seleccionarlo y cargar la información de sus alumnos a una asignatura.

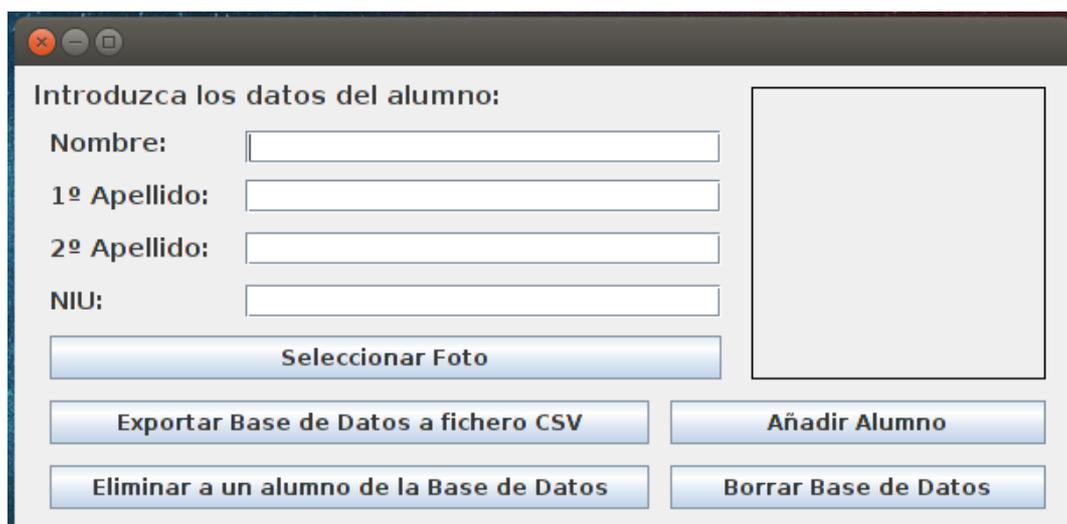


Figura 5.1.3. Interfaz de la aplicación Java de escritorio.

Desarrollo de la aplicación Android del alumno

Una vez desarrollada la aplicación explicada anteriormente, se ha comenzado a desarrollar la aplicación que utilizan los alumnos para generar sus propios marcadores. Esta aplicación utiliza la librería ArUco para Android para generar los submarcadores del marcador que se ha definido y que se muestra en la figura 5.1.2.4.

El primer paso de desarrollo de esta aplicación ha sido lograr que la librería para Android generara los marcadores simples. Una vez que la aplicación permitía generar marcadores entre 0 y 1023, se ha modificado el proceso de generación para crear los marcadores compuestos por cuatro submarcadores.

El proceso de creación ha consistido primero en generar los cuatro submarcadores haciendo llamadas a la librería y luego rotarlos dependiendo de los dos primeros dígitos del NIU como se ha explicado en la fase de diseño. Por último se dibujan los triángulos inferiores de color negro que se aprecian en la figura 5.1.2.4. Cuando el usuario abandona la pantalla donde se muestra el marcador que ha generado, se ha añadido la posibilidad de guardar dicho marcador para mostrarlo en otro momento sin necesidad de volverlo a generar. Esta funcionalidad puede ser útil en caso de que un alumno tenga que mostrar su marcador de forma recurrente y no desee tener que estar generándolo cada vez.

Desarrollo de la aplicación Android del profesor

Una vez desarrollada la aplicación del alumno, ha sido desarrollada la aplicación que utilizan los profesores para pasar lista. El desarrollo de esta aplicación se ha realizado a su vez en varias fases.

La primera fase ha consistido en lograr vincular la aplicación a la cuenta de Google Drive del profesor. Para lograr este objetivo se han tenido que consultar diversos tutoriales y descartar ideas por la imposibilidad de

llevarlas a cabo. Por ejemplo, una de las ideas que se han tenido que descartar ha sido la de utilizar una hoja de cálculo de Google Drive en la que cuando el profesor fijara la asistencia de un alumno desde la aplicación, se marcara automáticamente en la asistencia de ese alumno. El problema encontrado ha sido que para implementar esta función se necesita el API para hojas de cálculo de Google que, aunque cuenta con diferentes adaptaciones para Android realizadas por desarrolladores independientes como *Google SpreadSheet Lib for Android* [17], no se ha logrado que funcione correctamente. Por ese motivo, finalmente se ha optado por utilizar el API de Google Drive para Android. Este API permite descargar o cargar ficheros a la cuenta de Google Drive que esté vinculada al dispositivo Android.

La segunda fase ha consistido en incluir la librería de ArUco para Android y en comprobar su correcto funcionamiento. Esta fase se ha podido realizar en poco tiempo porque ya se había adquirido el conocimiento de cómo implementarla al desarrollar la aplicación del alumno. En esta aplicación la librería se utiliza para analizar las imágenes que captura la cámara y obtener la información relativa al marcador que se detecte.

La tercera fase ha consistido en implementar el funcionamiento de cámara del dispositivo Android. Para ello se ha utilizado OpenCV [18], una librería que permite hacer uso de un listener que gestiona cada frame obtenido desde la cámara. Este método realiza una llamada a la librería ArUco con el frame obtenido y comprueba si se detecta algún marcador. El marcador que se ha definido está compuesto por cuatro submarcadores, por lo que cuando en la imagen se detectan menos de cuatro submarcadores la única función de este método es pintarlos de diferentes colores.

La cuarta y última fase ha consistido en implementar la base de datos con la información de los alumnos para poder mostrarla cuando uno de ellos es reconocido. Como se ha comentado anteriormente, se ha utilizado el gestor de bases de datos H2 [13] en su adaptación para Android. Este gestor es

muy útil porque aparte de las capacidades comunes para crear, modificar o borrar tablas en bases de datos cuenta con funciones extra que han sido necesarias en el desarrollo de la aplicación. Estas funciones consisten en permitir importar una tabla desde un fichero CSV y exportar una tabla a un fichero CSV. Para usar estas funciones sólo se debe incluir CSVREAD o CSVWRITE en la sentencia SQL, de manera que, si por ejemplo queremos importar una tabla a la asignatura DISEÑO, se debe utilizar la sentencia:

```
CREATE TABLE DISEÑO AS SELECT * FROM CSVREAD ('fichero.csv');
```

Estas funciones facilitan que un profesor pueda importar los alumnos de una asignatura desde un fichero CSV de su cuenta de Google Drive y que pueda exportar las listas a un fichero CSV y subirlas a Google Drive.

5.1.4 Fase de testeo

Esta última fase ha consistido en probar el funcionamiento de las aplicaciones desarrolladas. Para ello se han realizado multitud de pruebas con diferentes bases de datos.

Gracias a esta última fase se han podido detectar y corregir errores entre los que se encuentran los siguientes:

- Error al identificar el alumno: se ha solucionado un error que hacía que los marcadores de los alumnos no se identificaran correctamente. El error estaba en que al decodificar el marcador y formar el NIU, había un par de dígitos que se colocaban en una posición errónea y eso provocaba el incorrecto funcionamiento de la aplicación.
- Error al guardar el marcador generado por un alumno: este error se producía porque la ruta que se había establecido para almacenar un marcador generado no era accesible desde la aplicación. Una vez modificada esa ruta a un directorio interno de la aplicación se solucionó el error.

- Error al exportar lista de asistencia a Google Drive: este error se producía porque se comenzaba a subir a Google Drive el fichero CSV con las listas antes de que dicho fichero se hubiera terminado de generar.
- Error al crear una asignatura con un nombre de varias palabras en la aplicación del profesor: este error se producía porque la sentencia para crear una tabla estaba mal escrita de manera que el gestor de bases de datos no entendía que el nombre podía contener espacios.

Esta fase de testeo ha sido una de las más importantes porque ha permitido solucionar problemas que no fueron detectados cuando se desarrolló el código.

Una vez finalizada esta fase, se han dedicado un par de días para crear una web [19] que permite compartir el resultado de este proyecto. En dicha web se pueden descargar las aplicaciones desarrolladas y acceder al código fuente de las mismas.



Figura 5.1.4. Captura de la web creada.

5.2 Funcionamiento de las aplicaciones Android

5.2.1 Funcionamiento Aplicación Alumno

La aplicación del alumno tiene un funcionamiento muy sencillo. Cuenta sólo con dos funciones: generar marcador y mostrar marcador guardado.

Cuando el usuario pulsa sobre generar marcador, se le abre un cuadro de diálogo que le pide el NIU necesario para generar el marcador. Una vez que el usuario introduce el NIU, la aplicación abre una nueva ventana en la que se muestra el marcador generado. Desde esa misma ventana el alumno puede mostrar el marcador a su profesor para que proceda a la identificación.

Cuando el usuario abandona esa ventana la aplicación muestra un nuevo cuadro de diálogo donde se le pregunta al usuario si desea guardar el marcador que acaba de generar. En caso de que el usuario responda afirmativamente, el marcador quedará almacenado en su dispositivo y el usuario podrá verlo de nuevo si pulsa sobre la opción de “Ver marcador Guardado”.

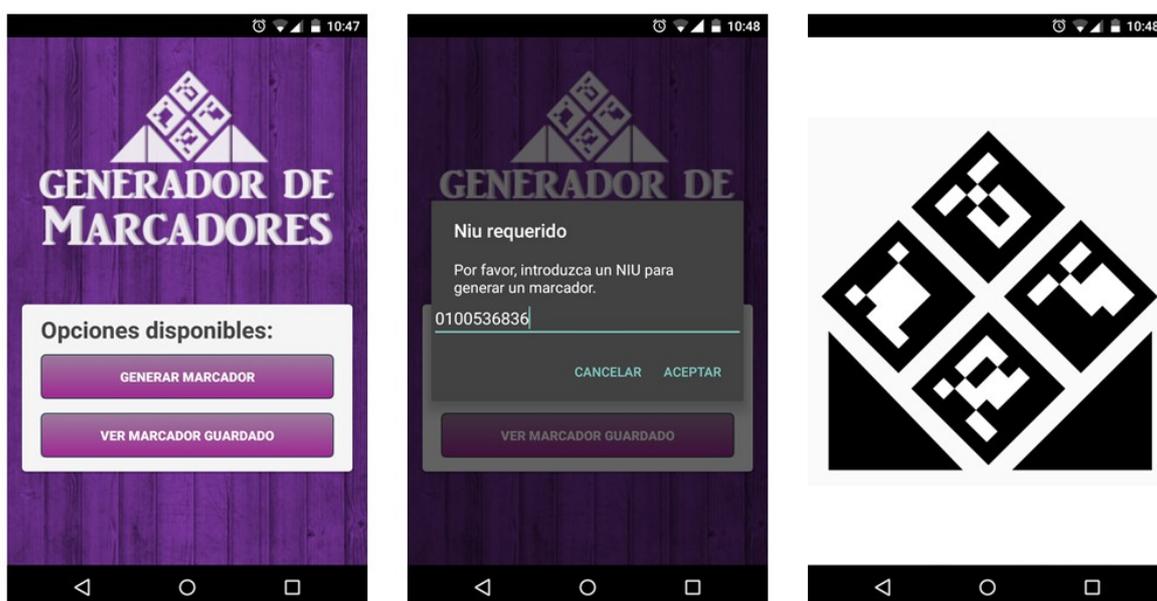


Figura 5.2.1. Capturas de la aplicación del alumno.

5.2.2 Funcionamiento Aplicación Profesor

La aplicación del profesor cuenta con una primera pantalla en la que se pide un nombre de usuario y contraseña. Estos datos se han establecido por defecto a “user” como nombre de usuario y “1234” como contraseña.

Una vez que se introducen estos datos, se abre la ventana principal de la aplicación en la que aparece la asignatura seleccionada y las diferentes opciones disponibles. Si la aplicación detecta que es la primera vez que se abre, se mostrará al usuario un cuadro de diálogo en el que se le informa que debe seleccionar un fichero CSV de su cuenta de Google Drive para continuar.



Figura 5.2.2.1. Pantalla de inicio de sesión y principal.

Una vez que el usuario seleccione un fichero y cree en la aplicación una asignatura podrá acceder a la pantalla principal. En esta pantalla se muestra el logo de la aplicación, la asignatura actual para la que se va a pasar lista y las diferentes opciones disponibles como iniciar reconocimiento de marcadores, importar una asignatura de Google Drive, exportar la listas a Google Drive y eliminar una asignatura del dispositivo.

En cuanto al bloque de información que muestra la asignatura actual, también se muestra un botón que permite modificar la asignatura sobre la que se va a pasar lista. Esta función ha sido desarrollada para un profesor que imparta clases en diferentes asignaturas y quiera realizar el control de asistencia utilizando la aplicación. Al pulsar sobre esta opción, se abrirá un cuadro de diálogo que contiene una lista con las diferentes asignaturas disponibles en el dispositivo Android. El profesor sólo debe pulsar sobre una de ellas para cambiar la asignatura actual. Para eliminar una de las asignaturas descargadas del dispositivo, el usuario debe pulsar sobre la opción “eliminar asignaturas” y pulsar sobre la deseada en la lista que se le muestra.

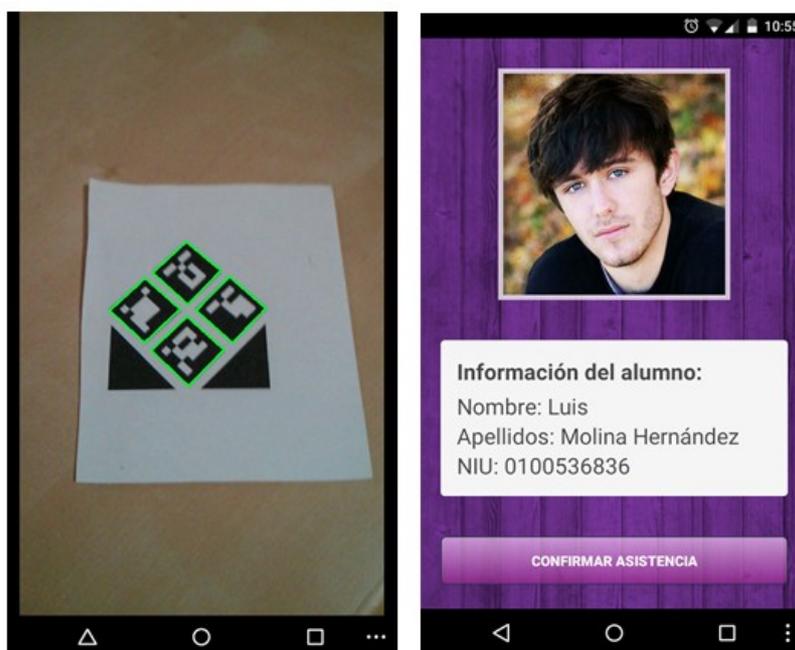


Figura 5.2.2.2. Pantalla de cámara y pantalla con la información de un alumno reconocido.

En cuanto a las opciones disponibles, la más importante es la de iniciar reconocimiento. Al seleccionarla, se abre una nueva ventana en la que se muestra la cámara. Desde esta ventana, el profesor debe escanear el marcador del alumno que desee identificar.

Cuando se reconoce un marcador y el NIU decodificado coincide con el NIU de algún alumno de la asignatura actual, la aplicación mostrará una nueva ventana con la información de dicho alumno. En esa ventana se mostrará su foto, nombre, apellidos y NIU. Si el profesor confirma que se trata del mismo alumno que le está mostrando el marcador pulsará sobre “confirmar asistencia”.

La aplicación automáticamente creará una columna en la tabla de la asignatura actual con la fecha del día (si no ha sido creada anteriormente) e indicará que ese alumno acudió a clase. En ese momento se volverá automáticamente a la ventana de la cámara para seguir reconociendo marcadores.

Por último, en la pantalla principal también están las opciones que permiten importar una asignatura y exportar las listas. En cuanto a la primera opción, cuando el usuario la selecciona, se abre una nueva ventana en la que se muestran todos los ficheros que contiene su cuenta de Google Drive. El usuario debe seleccionar el fichero CSV con los alumnos de la asignatura que desea importar. Una vez seleccionado, la aplicación abre un nuevo diálogo en el que el profesor deberá introducir el nombre de la asignatura que está importando. La aplicación utilizará ese nombre para crear una tabla en la base de datos con los alumnos que contiene el fichero seleccionado.

Si el usuario pulsa sobre “exportar base de datos”, la aplicación abrirá un nuevo cuadro de diálogo para que el profesor introduzca el nombre del fichero CSV que va a generar. Una vez introducido, la aplicación exportará la tabla de la asignatura actual a un fichero CSV y lo subirá a su cuenta de Google Drive. Una vez finalizado este proceso, el profesor puede consultar ese fichero y manejar la información que contiene como crea conveniente.

Capítulo 6

Conclusiones y líneas futuras

Este proyecto ha permitido desarrollar un sistema de control de asistencia que puede resultar útil a los profesores de la Universidad de La Laguna. Algunas de las ventajas del sistema desarrollado son:

- Comodidad y facilidad de uso: es un sistema bastante cómodo puesto que el profesor sólo debe preocuparse de llevar consigo su dispositivo móvil con Android. Además, el proceso de marcar la asistencia de un alumno es muy sencillo de realizar, primero se escanea su marcador y luego se marca que asiste.
- Coste muy bajo de implementación: dado que este sistema utiliza tecnologías ya existentes, el coste de implementación puede llegar a ser nulo. Sólo en el caso de que el profesor no cuente con un dispositivo Android habría que considerar los gastos de adquirir uno. Si el alumno no cuenta con un dispositivo Android no es problema, porque tiene la posibilidad de pedirle a otro compañero que genere su marcador y se lo envíe a su dispositivo móvil. Además, en caso de no contar con uno, podría imprimirlo en papel y mostrar así su marcador al profesor.

En cuanto a los inconvenientes de este sistema, principalmente podemos encontrar uno:

- Proceso de introducción de alumnos en la base de datos: cuando el profesor utilice la aplicación de escritorio para generar la base de datos

con los alumnos de su asignatura, puede ser que se encuentre con un proceso tedioso si cuenta con muchos alumnos. Este inconveniente se podrá solventar en el futuro si se vincula la aplicación de escritorio con el campus virtual, de manera que el profesor pueda descargar un fichero CSV directamente que contenga la información de sus alumnos.

En cuanto a las líneas futuras de trabajo, este proyecto tiene muchas funciones que se pueden implementar. Algunas de estas nuevas funciones o aspectos son:

- Versiones para IOS y Windows Phone de las aplicaciones del profesor y alumno.
- Vinculación del campus virtual de la universidad con la aplicación que genera la base de datos con los alumnos de una asignatura. Esto permitiría para agilizar dicho proceso de generación.
- Añadir una función a la aplicación del alumno para que pueda compartir un marcador generado o imprimirlo directamente desde el dispositivo móvil.
- Añadir una función a la aplicación del profesor para que se envíe un correo electrónico a los alumnos que han faltado a clase indicando cuantas faltas llevan acumuladas.
- Añadir una función a la aplicación del profesor para que pueda modificar directamente desde el dispositivo móvil la información que tiene almacenada de un alumno. Esta función permitiría corregir errores que hubiera cometido al introducir los datos o modificar la foto del alumno por una tomada en ese momento.

Capítulo 7

Summary and Conclusions

In this project we have developed an assistance control system that can be very useful for the teachers of the University of La Laguna. Some of the advantages of this system are:

- Easy to use: this is a very easy to use system because the teacher only have to worry about carrying his Android mobile phone. Furthermore, the process of setting the assistance of a student is very easy, first the teacher scans his marker and then he set the student assistance.
- Very low implementation cost: this system makes use of already existing technologies so the implementation cost are almost zero. We should only invest some money if the teacher does not have an Android mobile device. There is no problem if the student does not have an Android mobile device because he can ask to other student to generate his marker and then he can show it as an image in his mobile phone. He could also print his/her marker in paper to show it to the teacher.

In the other hand, there is one disadvantage of this system:

- Students introduction process in the database: when the teacher uses the desktop application to generate the database with all the students of his subject, it may take a lot of time if he has many students. This disadvantage can be solved in the future if we link the desktop application with the university online platform. This way, the teacher will be able to download directly a CSV file with the information of his

students.

This project has some functions not yet implemented to develop in the future. Some of these functions are:

- IOS and Windows Phone version of the teacher and student application.
- Link university virtual campus with the desktop application which generates the students database.
- Add one function to the student application for sharing a generated marker or for printing directly from the mobile device.
- Add one function to the teacher application for sending an e-mail to the students who do not came to class. Add also information about how many absences have they.
- Add one function to the teacher application for allowing the modification of the student information using the mobile device. This function could be very helpful to correct mistakes in the student information and to change the student photo with other taken with the mobile device.

Capítulo 8

Presupuesto

Para crear el presupuesto se ha tenido en cuenta el tiempo dedicado a cada una de las fases de desarrollo y una cantidad estimada del costo del trabajo por hora.

8.1 Presupuesto por fases

Fase	Euros/Hora	Horas trabajadas	Total
Documentación	35 €/h	80 h	2800 €
Diseño	35 €/h	100 h	3500 €
Desarrollo	40 €/h	160 h	6400 €
Testeo	35 €/h	40 h	1400 €
TOTAL			14100 €

Tabla 8.1: Presupuesto.

Bibliografía y referencias

[1] Fiducial. http://es.wikipedia.org/wiki/Marcador_de_referencia [On-line, consultado en junio de 2015]

[2] CSV. <http://es.wikipedia.org/wiki/CSV> [On-line, consultado en junio de 2015]

[3] SGD App <http://www.tecnausa.com/educacion/sgd-android-app.html> [On-line, consultado en junio de 2015]

[4] Asistente Docente <https://play.google.com/store/apps/details?id=com.ddroid.portafoliodocente&hl=es> [On-line, consultado en junio de 2015]

[5] Cuaderno del Profesor <https://play.google.com/store/apps/details?id=com.labeldroid.cuaderno.pro&hl=es> [On-line, consultado en junio de 2015]

[6] Cuaderno del Profesor Additio <https://play.google.com/store/apps/details?id=com.additioapp.additio&hl=es> [On-line, consultado en junio de 2015]

[7] Teacher Kit <https://itunes.apple.com/us/app/teacherpal/id389584618?mt=8#> [On-line, consultado en junio de 2015]

[8] iTeacherBook <https://itunes.apple.com/es/app/iteacherbook/id459153342?mt=8> [On-line, consultado en junio de 2015]

[9] iDoceo <https://itunes.apple.com/es/app/idoceo-teachers-notepad/id477120941> [On-line, consultado en junio de 2015]

[10] Vía-Móvil <http://www.via-movil.es/> [On-line, consultado en junio de 2015]

[11] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, M.J. Marín-Jiménez, Automatic generation and detection of highly reliable fiducial markers under occlusion, Pattern Recognition, Volume 47, Issue 6, June

2014, Pages 2280-2292, ISSN 0031-3203,

<http://dx.doi.org/10.1016/j.patcog.2014.01.005>.

(<http://www.sciencedirect.com/science/article/pii/S0031320314000235>)

[12] Fiala M., Designing Highly Reliable Fiducial Markers, Pattern Analysis and Machine Intelligence, IEEE Transactions Volume 32, Issue 7, July 2010, Pages 1317 – 1324, ISSN 0162-8828,

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5184844&isnumber=5466012>

[13] H2 <http://www.h2database.com/html/main.html> [On-line, consultado en junio de 2015]

[14] MarvelApp <https://marvelapp.com/> [On-line, consultado en junio de 2015]

[15] ArUco <http://www.uco.es/investiga/grupos/ava/node/26> [On-line, consultado en junio de 2015]

[16] Base64 <https://en.wikipedia.org/wiki/Base64> [On-line, consultado en junio de 2015]

[17] Google SpreadSheet Lib for Android <https://code.google.com/p/google-spreadsheet-lib-android/> [On-line, consultado en junio de 2015]

[18] OpenCV <http://opencv.org/about.html> [On-line, consultado en junio de 2015]

[19] Web de descarga asistenciav1.hol.es [On-line, consultado en junio de 2015]

[20] Método Otsu https://es.wikipedia.org/wiki/M%C3%A9todo_de_valor_umbral#M%C3%A9todo_de_otsu [On-line, consultado en junio de 2015]